

*Accredited by NBA - AICTE and Accredited by NAAC - UGC with 'A++' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai*

## Department of Computer Applications

Course Code: 23CAT606

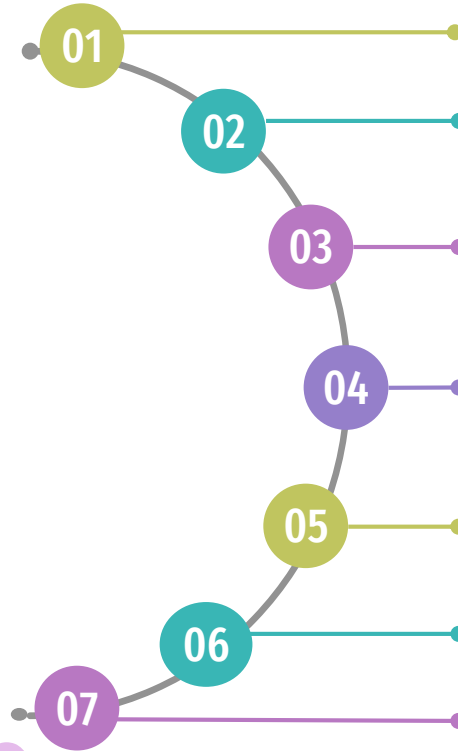
Course Name: Java Programming

Unit I: Java Fundamentals

Topic 3: Class and object



# Java Features



- Package
- Java Program Structure
- Data type
- Operators
- Control Statements
- Looping
- Conditional statements

# Class

A class is a group of objects which have common properties.

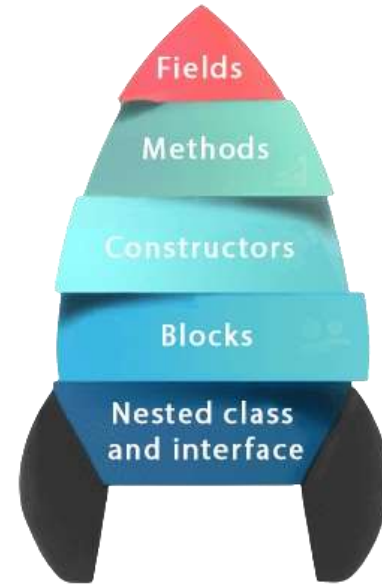
It is a template or blueprint from which objects are created.

## Syntax to declare a class:

```
class <class_name>{
    field;
    method;
}
```

## Instance variable in Java

A variable which is created inside the class but outside the method is known as an instance variable.



# Method in Java



In Java, a method is like a function which is used to expose the behavior of an object.

## Advantage of Method

1. Code Reusability
2. Code Optimization

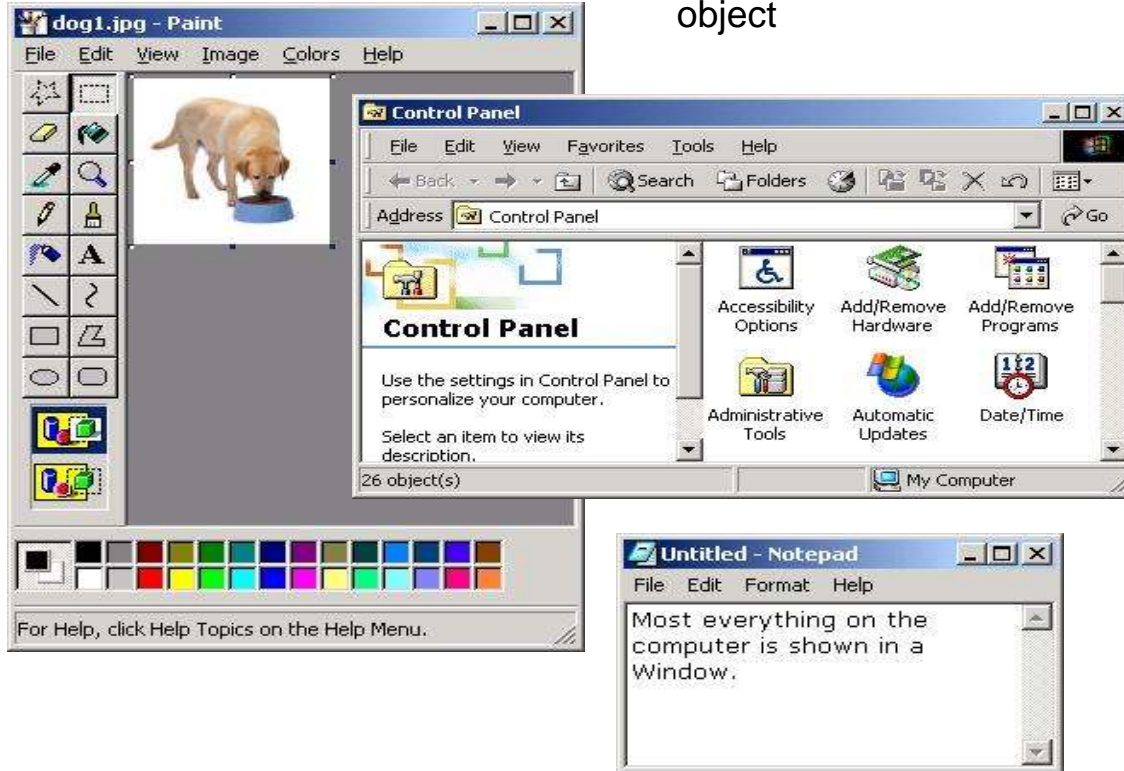
## new keyword in Java

The new keyword is used to allocate memory at runtime. All objects get memory in Heap memory area.

# What is an object in Java



An entity that has state and behavior is known as an object



## Objects: Real World Examples



# Three characteristics of Object

**A** **State**  
Represents the data of an object.

**B** **Behavior**  
represents the behavior of an object such as deposit, withdraw, etc.

**C** **Identity**  
It is used internally by the JVM to identify each object uniquely.

# Object Definitions



**An object is an instance of a class.** A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

1. An object is a real-world entity.
2. An object is a runtime entity.
3. The object is an entity which has state and behavior.
4. The object is an instance of a class

## Object and Class Example: main within the class

```
class Student
{
    int id;
    String name;
    public static void main(String args[])
    {
        Student s1=new Student();
        System.out.println(s1.id);
        System.out.println(s1.name);
    }
}
```

Output: 0 null

# Object and Class Example: main outside the class



```
class Student{
    int id;
    String name;
}
class TestStudent1{
    public static void main(String args[]){
        Student s1=new Student();
        System.out.println(s1.id);
        System.out.println(s1.name);
    }
}
```

0 null



# 3 Ways to initialize object



1. By reference variable
2. By method
3. By constructor

## 1. By reference variable

```
class Student{
    int id;
    String name;
}
class TestStudent2{
    public static void main(String args[]){
        Student s1=new Student();
        s1.id=101;
        s1.name="Sonoo";
        System.out.println(s1.id+" "+s1.name);
    }
}
```

## 2. By method

```
class Student{
    int rollno;
    String name;
    void insertRecord(int r, String n){
        rollno=r;
        name=n;
    }
    void displayInformation(){
        System.out.println(rollno+" "+name);}
}
class TestStudent4{
    public static void main(String args[]){
        Student s1=new Student();
        Student s2=new Student();
        s1.insertRecord(111,"Karan");
        s2.insertRecord(222,"Aryan");
        s1.displayInformation();
        s2.displayInformation();
    }
}
```

# 3 Ways to initialize object



1. By reference variable
2. By method
3. By constructor

## 3. By Constructor

```
class Bike1{  
    Bike1()  
    {  
        System.out.println("Bike is created");  
    }  
  
    public static void main(String args[]){  
        Bike1 b=new Bike1();  
    }  
}
```

# Anonymous object



Anonymous simply means nameless. An object which has no reference is known as an anonymous object. It can be used at the time of object creation only.

```
new Calculation();
```

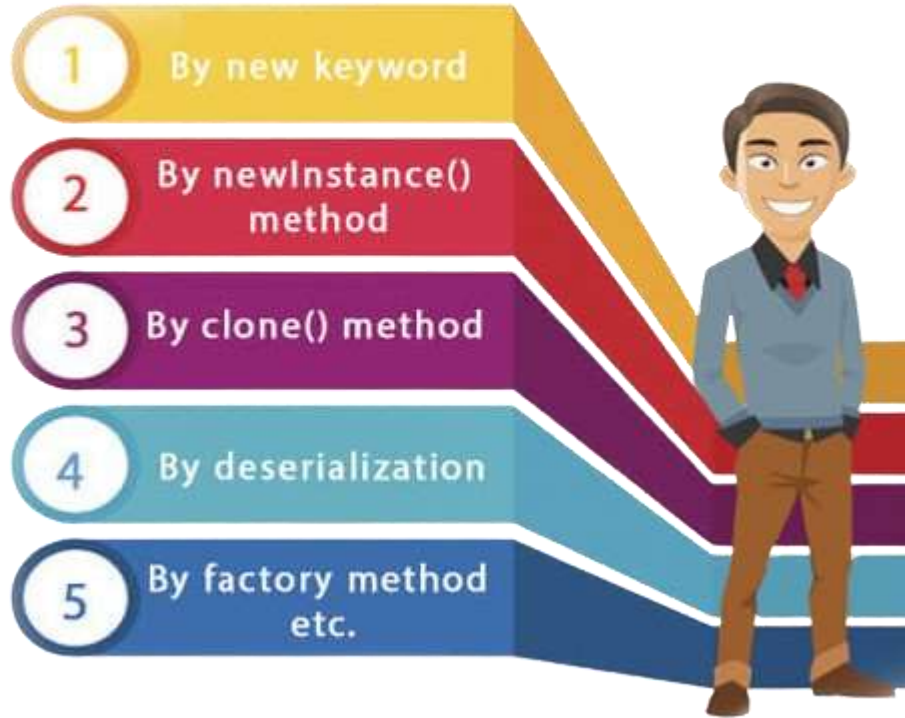
## Creating multiple objects by one type only

```
Rectangle r1=new Rectangle(), r2=new Rectangle();
```

```
class Calculation{
    void fact(int n){
        int fact=1;
        for(int i=1;i<=n;i++){
            fact=fact*i;
        }
        System.out.println("factorial is "+fact);
    }
}

public static void main(String args[]){
    new Calculation().fact(5);
}
}
```

# Different way to create a object



# Summary

---

①

②

③

